A Self-Learning Repeated Game Framework for Optimizing Packet Forwarding Networks

Zhu Han, Charles Pandana, and K.J. Ray Liu
Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742.

***Abstract*— For the networks with packet forwarding, distributed control to enforce cooperation for nodes' packet forwarding probabilities is essential to maintain the connectivity of networks. In this paper, we propose a novel self-learning repeated game framework to optimize packet forwarding probabilities of distributed users. The framework has two major steps: First, an adaptive repeated game scheme ensures the cooperation among users for the current cooperative packet forwarding probabilities. Second, a self-learning scheme tries to find better cooperation probabilities. Some special cases are analyzed to evaluate the proposed framework. From the simulation results, the proposed framework demonstrates the near optimal solutions in both symmetry and asymmetry network.**

I. Introduction

Recent advances in wireless communication have made possible the large-scale deployment of wireless ad-hoc and sensor networks, which consist of small, low-cost nodes with simple processing and networking capabilities. The nodes in the scenarios mentioned above may not be willing to fully cooperate. And in fact, it is reasonable to assume that the nodes are selfishly maximizing their own benefits for the following reason. From the node perspective, forwarding the arriving packets consumes its limited battery power. Therefore, it may not be of the node's interest to forward all the arriving packets. In contrast, reject forwarding other's packets will adversely affect the network connectivity. Hence, it is very crucial to design a distributed control mechanism that enforces cooperation among participating nodes.

Since the distributed nodes don't have information of others, they act selfishly to optimize their own performances. This motivates us to apply Game Theory approach [1] to packet forwarding problem. In [2], repeated game theory was applied to routing problems. In [3], multiple access resource allocation was studied using game theory approach. In [4], repeated game was further applied to physical layer problems. The distributed control mechanism enforces collaboration/cooperation has also been studied within the context of game theory in several existing literatures. Srinivasan et al. [5] provide a mathematical framework for cooperation in ad-hoc networks, which focuses on the energy-efficient aspects of cooperation. In [6], the authors focus on the properties of the cooperation enforcement mechanisms used to detect and prevent selfish behavior of nodes in ad-hoc network. They show that the formation of large coalitions of cooperating nodes is possible when mechanism similar to CORE [7] is used. In [8], the authors consider a less aggressive punishment policy. In this scheme, the node uses the minimum forwarding probability among its neighborhood as its forwarding probability after detecting the misbehavior. Felegyhazi et al. [9] considers a model to show cooperation among participating nodes and provides sufficient condition on the network topology under which each node employing the punishment strategy results in a Nash Equilibrium. In [7] and [10], the authors define protocols that are based on a reputation system. In [11], the authors propose a repeated game framework for multiple-access using Cartel maintenance. Other work of applying cooperative game theory and noncooperative game theory to OFDMA networks can be found in [12] [13].

In this paper, we propose a self-learning repeated game framework for users to obtain the optimal packet forwarding probabilities distributively. The framework has two major steps. First, to ensure cooperation among users, the users apply adaptive repeated game scheme to punish the greedy users from deviation and play non-cooperatively. Second, the users try to learn the better packet forwarding probabilities that generate better performances. From the simulation results, the proposed scheme can find the optimal solutions or near optimal solutions in both symmetry network and asymmetry network.

The proposed scheme has an analogy to the human society. Before the civilization, there were no rules in the society to enforce cooperation. People fought each other greedily and non-cooperatively. The consequences were low social productivity and low living standards for the people themselves. Then through revolutions, new relationships among human beings were proposed such as slavery, feudalism, and capitalism, etc. In order to maintain the new relationship, rules such as laws are defined to enforce people for cooperation under the new relationship. Similarly in packet forwarding network, it has been proved from previous works that the network performance will degrade to zero asymptotically if no cooperation is enforced. If we can enforce the cooperation among distributed and greedy users and if we can find the better relationship that users forward others' packets, the system efficiency as well as all users' performances can be improved.

This paper is organized as follows: In Section II, the system model is given and the problem is formulated. In Section III, we proposed the self-learning repeated game framework for packet forwarding networks. In Section IV, we analyze some special cases to evaluate the performance of the proposed scheme. In Section V, conclusions are drawn.

II. System Model and Problem Formulation

In packet forwarding networks, packet forwarding problem is essential for distributed users to get connected to the destinations. Suppose there are a total of K users. The k^{th} user has a total of N_k routes for its packet transmission. In this paper, we assume the routes have been determined and

 0-7803-8966-2/05/$20.00

known. Let's define I_k^i as the set of the nodes on the i^{th} route for the k^{th} user. Suppose each user has the willingness to forward other user's packet with probability of α_i. For each user, the successful transmission or reception of one packet will have the benefit of G and forwarding others' packet will cost F per packet. Suppose the k^{th} user transmits its packet with probability of P_k^i to the i^{th} route. Obviously, we have $\sum_{i=1}^{N_k} P_k^i = 1$. So the utility function U_k for the k^{th} user can be expressed as:

$$U_k = \sum_{i=1}^{N_k} P_k^i G \Pi(\alpha_j, j \in I_k^i) - F\alpha_k B_k, \quad (1)$$

where Π is the successful transmission probability which is a function of packet forwarding probabilities along the routes. B_k is the forward request probability from other users. The first term on the right hand side of above equation is the average benefit for the k^{th} user, which depends on other users' willingness for forwarding. The second term means the cost of forwarding other users' packet which depends on its own willingness for packet forwarding.

Since an individual user in networks has less information about others and may selfishly optimize its own performance, the packet forwarding problem is, in some sense, analogous to the economy system of the human society. Game theory is a successful economy approach, which studies the mathematical models of conflict and cooperation between intelligent and rational decision makers. In this paper, we can formulate this problem as a noncooperative game problem where each user adjusts its forward probability to maximize its own utility function:

$$\max_{0 \le \alpha_k \le 1} U_k(\alpha_k, \alpha_{-k}) \quad (2)$$

where $\alpha_{-k} = [\alpha_1, \ldots, \alpha_{k-1}, \alpha_{k+1}, \alpha_K]^T$ is the other users' behaviors of packet forwarding. To analyze the outcome of the game, Nash Equilibrium is a well-known concept, which states that in equilibrium every user will select a utility-maximizing strategy given the strategy of every other user.

Definition 1: Define feasible range Ω as $[0, 1]$. Nash Equilibrium $[\hat{\alpha}_1, \ldots, \hat{\alpha}_K]^T$ is defined as:

$$U_k(\hat{\alpha}_k, \alpha_{-k}) \le U_k(\tilde{\alpha}_k, \alpha_{-k}),\ \forall k,\ \forall \tilde{\alpha}_k \in \Omega,\ \alpha_{-k} \in \Omega^{K-1}. \quad (3)$$

i.e., given the other users' packet forward probability, no user can increase its utility alone by changing its own packet forward probability.

Unfortunately, the Nash equilibrium for the packet forwarding game in (3) is usually $\hat{\alpha}_k = 0,\ \forall k$, because each user's benefit depends on other users' willingness for forwarding and does not depend on its own behavior, while the user's cost solely depends on its willingness for packet forwarding. So each user will greedily drop its packet forwarding probability to reduce the cost and increase the utility. However, if all users don't forward, the successful packet transmission probabilities might become zero. Consequently the benefits for users are zeros and the whole system turns down. So if the users play noncooperatively and have the Nash equilibrium, all users' utility might be zero. On the other hand, if the users can cooperate and have some positive packet forwarding probability, all users can have benefits.

So the problem can be formulated as to design a method to enforce cooperation among users in packet forwarding. First, we want to find the best packet forwarding vector such that the utilities of all users are strictly better than those of Nash equilibrium. Moreover, we want to design a mechanism to enforce such cooperation among users. Since this problem is very similar to some problems in the human society, in the next section, we use the economy approach called repeated game to solve the proposed problem.

III. Self-learning Repeated Game Framework

A. Repeated Game Basic

In order to solve the proposed problem, we apply the repeated game approach to packet forwarding problem. Repeated game is a special case of dynamic game (a game that is played multiple times). When players interact by playing a similar static game which is played only once like (2) numerous times, the game is called a repeated game. Unlike a game played once, a repeated game allows a strategy to be contingent on past moves, thus allowing reputation effects and retribution, which give possibility for cooperation.

Definition 2: For T-period *repeated game*, at each period t, the moves during periods $1, \ldots, t-1$ are known to every player. β is the discount factor. The total discounted payoff for each player is computed by

$$\sum_{t=1}^{T} \beta^{t-1} U_k(t) \quad (4)$$

where $U_k(t)$ denotes the payoff to player k in period t. If $T = \infty$, the game is referred as the infinitely-repeated game. The average payoff to player k is then given by:

$$\bar{U}_k = (1-\beta) \sum_{t=1}^{\infty} \beta^{t-1} U_k(t) \quad (5)$$

From the game theory literature, the repeated game can enforce the greedy user to show cooperation. This is because the user will get punishment from other users in the near future if it acts greedily. The benefit of greediness will be eliminated by the loss of punishment in the future. So the users would rather act cooperatively. So the remaining problem is how to define a good rule to enforce the cooperation. From Folk Theorem, we know that in an infinitely repeated game, any feasible outcome that gives each player better payoff than the Nash equilibrium can be obtained.

Theorem 1: Folk Theorem [1]: Let $(\hat{\alpha}_1, \ldots, \hat{\alpha}_n)$ be the payoffs from a Nash equilibrium of G and let $(\alpha_1, \ldots, \alpha_n)$ be any feasible payoffs. There exists an equilibrium of the infinitely repeated game that attains $(\alpha_1, \ldots, \alpha_n)$ for $\alpha_i > \hat{\alpha}_i, \forall i$ as the average payoff, provided that β is sufficiently close to 1.

Now, we know that by using the repeated game, the greedy users can be forced to cooperate and have better payoffs. So the remaining problem is defining a good mechanism to enforce the cooperation, which we will show in the next subsection.

B. Self-learning Cooperation Enforcing Framework

The basic idea for the proposed algorithm is to let distributed users learn the optimal packet transmission probability step by step, while within each step, the strategy of repeated game is applied to ensure the cooperation among users. For simplicity, we omit the user index in this subsection. The block diagram of the algorithm is shown in Figure 1 and detail descriptions are as follows:

During **Initialization**, all users play noncooperative game and all user are balanced in an inefficient Nash equilibrium $\hat{\alpha}$. We set the time counter $n = 0$, the punishment time $T = 0$, and trigger threshold $V = \hat{\alpha}$.

In the next step, we play **repeated game** strategy. If all users play cooperatively, every user will have some benefits. However, from (1), if any user deviates from cooperation by playing noncooperatively and other users still play cooperatively, this user will have more benefits, while others suffer with lower benefits because of this user's greediness. In order to prevent users from deviation, the repeated game strategy provides a punishment mechanism. The basic idea is that each user sees if the utility function is lower than the threshold V. If so, that means some user may have deviated, then this user also plays noncooperatively for a period of time T. By doing this, the greedy user's short term benefit will be eliminated by the long term punishment. If all users concern the long term payoff such as (5), which is true by the assumption of rational users, then all users will have no incentive to deviate from cooperation.

In details, the repeated game scheme with parameter (V, T), for all users is explained as follows: Each user's utility U is compared with the threshold V. If $U < V$, i.e., someone deviates, the time counter n is set to zero, punish time is increased by one, and the user plays noncooperatively for a T period of time. Since we assume all users are rational, with increasing of T, the benefit of one time deviation will be eliminated out sooner or later. So finally, no user wants to deviate and $U \geq V$. At this time, the counter n starts increasing. If the system is stable in the cooperation for a period of time N, where N is a predefined constant, the algorithm assumes that the cooperation is enforced, and changes to the next step to improve the current cooperation.

In the next step, the algorithm tries to **self-learn** the optimal forward probabilities by modifying α with the goal to optimize the performances. The simplest way is to randomly generate $\alpha \in [0, 1]$, where different users may have different α. In the next time slot, all users observe if their performances become better. If not, the α is changed to the previous value. Otherwise, each user selects its packet forwarding probability as α, updates its threshold to current benefit $V = U$, calculates the difference of cooperation and noncooperation for the utility ΔU as

$$\Delta U = U(\text{new } \alpha) - U(\hat{\alpha}), \tag{6}$$

and calculates the deviation benefit ΔD. If the network is symmetry, the optimal punishment time can be written as:

$$T = \frac{\Delta D}{\Delta U}, \tag{7}$$

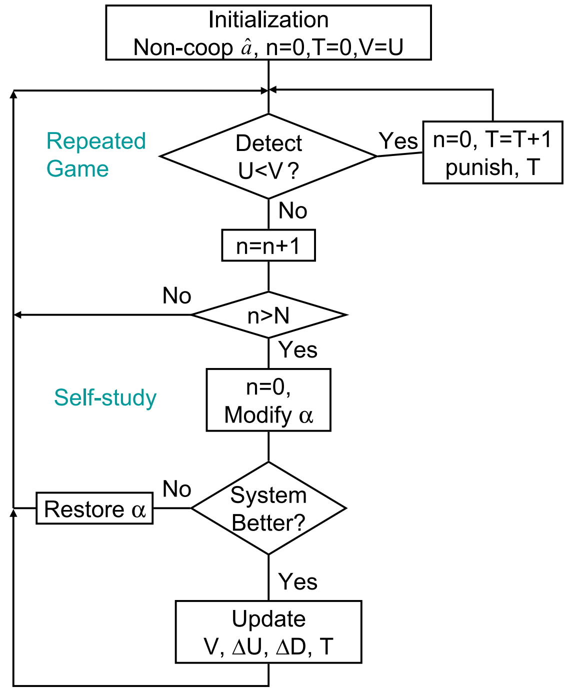

Fig. 1: Proposed self-learning repeated game framework

where T is the estimated punish time that prevents the users from deviation. Then the algorithm goes back to the repeated game case to update the punishment time T such that all users are willing to cooperate.

Notice that during the first time slot after α is modified, all users will act cooperatively, because of the rationale that deviation eliminates the chance of utility improvement in the future. In the repeated game step, the benefit of instantaneous deviation is eliminated sooner or later as long as the discount factor β is close enough to 1. So the T will converge to some value. In the self-learning step, if the new sets of α are not good for all users, the original value of α will be restored. If the new sets of α are good, the cooperation can be enforced by the future repeated game step. So the framework will converge.

In summary, the framework uses the threat of punishment to maintain the cooperation for the current α and try to learn if there is a better α for cooperation. In the next section, we will give some cases to analysis and evaluate the behaviors of the proposed framework.

C. Discussions of Asynchronous Networks

In the previous analysis, we assume the networks are synchronous, i.e., each user's utility can be observed instantaneously whenever other users deviate. This might not be true in the real networks. In this subsection, we will discuss the problem introduced by asynchronous networks and some possible solutions.

When the network is asynchronous, the deviation of users will be detected by other users with some time delay. This is not the problem. The problem is when the punishment period is over, the users may return to the cooperation phase in different time. This may trigger some users to continue punishment because they cannot distinguish if the users are deviating or the users are still in the punishment phase. The

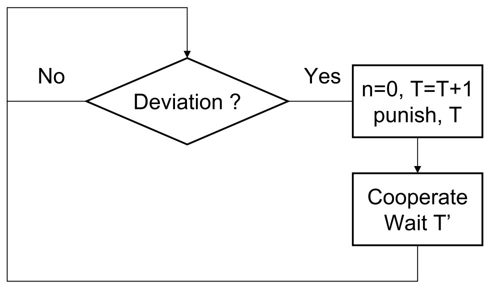

Fig. 2: Modified Repeated Game Step

problem will make the network fluctuating and the punishment time T cannot converge. In order to solve such a problem, we propose the following modification to the repeated game step of the proposed framework.

The modified repeated game step is shown in Figure 2, where an extra step is added at the end of punishment period. After switching back to cooperation, the user will wait for time T' and then observe if others deviate. This time T' is reserved for the other users to return to cooperation. This value is determined by the scale and topology of the networks. If the value of T' is too small, the network will not be stable and punishment period is always prevail because some users' delayed return to cooperation triggers others' new punishment periods. If the value of T' is too large, it gives the opportunities for greedy users to deviate to gain the benefits without detecting from other users. There are tradeoffs for selection of the value for T'.

The other concern is during the step when α is modified after the system is stable $n > N$. The message for all users to modify α can be implemented by the protocol like flooding. This message will take times to arrive every node. So to judge if the system becomes better, a user needs to wait for a period of time that may be similar to T'.

IV. Cases Analysis and Performance Evaluations

In the following subsections, we analyze two cases: symmetry networks and asymmetry networks. Some simple examples are given and analytical optimal results are deduced. Simulation results of the proposed framework are conducted to evaluate the performances.

A. Symmetry Networks

First we analyze the characteristic of the symmetric network. The topology of such a network is symmetry, consequently, the resulting Nash equilibrium and the optimum of the packet forwarding probabilities should be the same for all users, i.e. $\hat{\alpha}_k = \hat{\alpha}_j,\ \alpha_k = \alpha_j,\ \forall k, j$. In general, the networks are asymmetry. However at the edges of networks where some nodes may equally access the networks, symmetry topology may exist and symmetry analysis can be applied.

In this subsection, we give an example on the analysis of the synchronous symmetry networks. Suppose the considered network is shown in Figure 3. In this network, there are six fixed routes: $1 \rightleftarrows 4$, $2 \rightleftarrows 5$, and $3 \rightleftarrows 6$. All the destinations are 3 hops away from the source. We consider the node's utility function as the reward obtained from successfully transmitting or receiving a packet. We also assume the forwarding others' packets consume resources such as energy, therefore forwarding contributes a cost (negative reward) to the utility

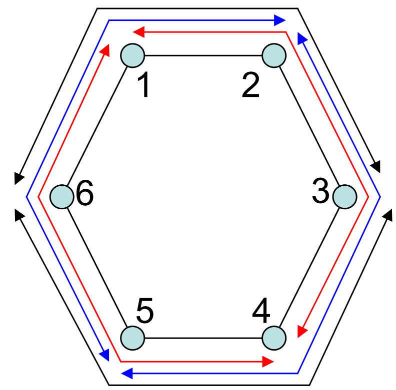

Fig. 3: Example of symmetric network

function. The utility functions for each of the node in Figure 3 are represented as follow:

$$\begin{aligned}
U_1 &= 2G[1-(1-\alpha_2\alpha_3)(1-\alpha_5\alpha_6)] - F[\alpha_1+\alpha_1\alpha_2]\\
U_2 &= 2G[1-(1-\alpha_1\alpha_6)(1-\alpha_3\alpha_4)] - F[\alpha_2+\alpha_2\alpha_3]\\
U_3 &= 2G[1-(1-\alpha_1\alpha_2)(1-\alpha_4\alpha_5)] - F[\alpha_3+\alpha_3\alpha_4]\\
U_4 &= 2G[1-(1-\alpha_2\alpha_3)(1-\alpha_5\alpha_6)] - F[\alpha_4+\alpha_4\alpha_5]\\
U_5 &= 2G[1-(1-\alpha_1\alpha_6)(1-\alpha_3\alpha_4)] - F[\alpha_5+\alpha_5\alpha_6]\\
U_6 &= 2G[1-(1-\alpha_1\alpha_2)(1-\alpha_4\alpha_5)] - F[\alpha_6+\alpha_1\alpha_6]
\end{aligned}$$

where α_i is the probability that node i is willing to forward others' packets, G is the reward for successfully transmitting and receiving a packet, and F is the cost for forwarding others' packet. We also assume that nodes are greedy and rational but not malicious, that is every node decides its forwarding probability to maximize its own utility function. If we consider the Nash equilibrium obtained noncooperatively from (2), obviously, to the best of each node's interest, every node selects zero forwarding probability (i.e. $\alpha_k = 0,\ \forall k$) to minimize its forwarding cost in the utility function. However, the overall network becomes disconnected as all the nodes act in noncooperative manner.

Note that due to the symmetry property of the network in Figure 3, the optimal forwarding probability and the corresponding utility of each node will be the same. We omit the subscript for simplicity. Consider the *system-wide* optimal solution to maximize everybody's utility, we can formulate the problem as:

$$\begin{aligned}
\max_{\alpha} \quad & U = 2G(2\alpha^2 - \alpha^4) - F(\alpha + \alpha^2)\\
\text{s.t.} \quad & 0 \le \alpha \le 1.
\end{aligned} \tag{8}$$

By differentiating the above equation, we obtain

$$\frac{\partial U}{\partial \alpha} = 8G(\alpha - \alpha^3) - F(1 + 2\alpha) = 0, \tag{9}$$

$$\alpha^3 - \left(1 - \frac{F}{4G}\right)\alpha + \frac{F}{8G} = 0. \tag{10}$$

The optimal forwarding probability in the symmetry network can be obtained by solving (10). Figure 4 shows the effects of forwarding probability α on the utility function for different normalized forwarding costs, F/G. We also show the optimal forwarding probabilities for different cases. It is obvious that as the cost for forwarding, F is smaller compared

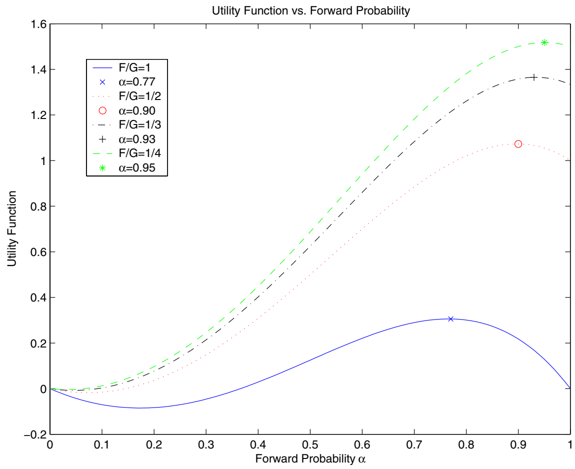

Fig. 4: Effect of forwarding probability to utility

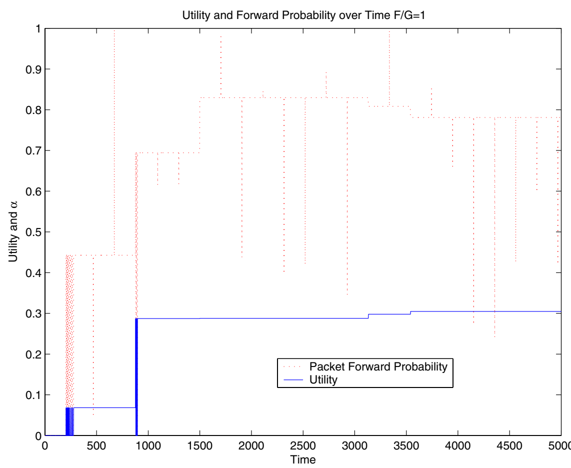

Fig. 5: Utility function and forwarding probability over time

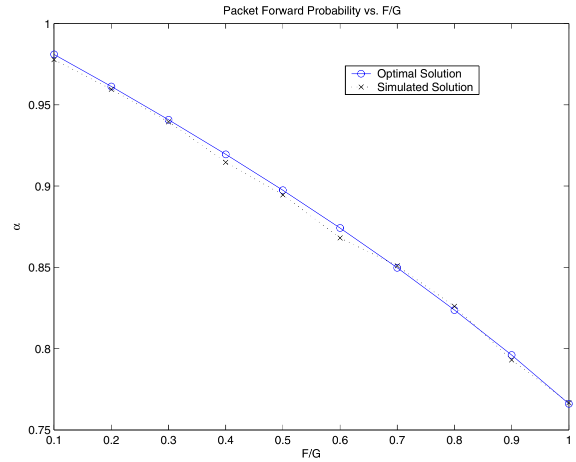

Fig. 6: Effect of F/G to optimal forwarding probability

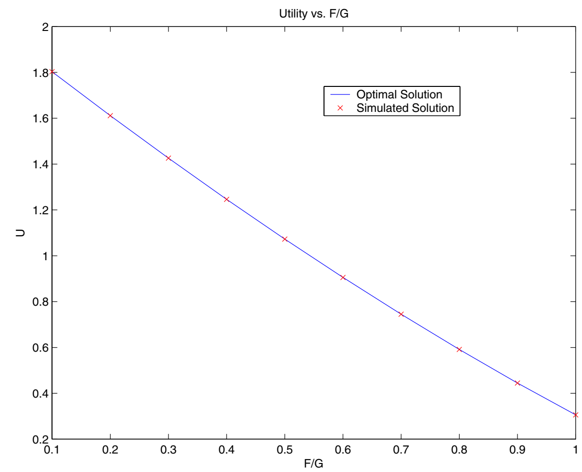

Fig. 7: Effect of forwarding cost to the optimal utility

to the transmitting/receiving reward, G, the optimal forwarding probability will approach the unity forwarding probability and the corresponding utility is also high. On the other hand, when F/G is large then every node has lower incentive to forward the others' packet and utility is low. This phenomenon is reasonable since when the cost for forwarding is very large, it is better for the node to save the energy for its own transmission. The goal of the cooperation mechanism design is to design the incentive for the nodes to avoid the noncooperative solution and to result in the *system-wide* optimal forwarding probability. It also worthy to mention that not every positive packet forwarding probability will generate the larger utility than full noncooperation case where the packet forwarding probability is zero. For example, when $F/G = 1$, the utility is higher than 1 only when $\alpha \geq 0.37$. So in the self-learning step, if α is modified less than 0.37, the system will have worse performance than noncooperation. As a result, the new α will be discarded and the original α is restored.

In Figure 5, we show the simulation results of the proposed framework for utility and packet forwarding probability over time. Here $F/G = 1$ and $N = 200$. Initially, $\alpha = 0$, because of the noncooperation transmission. Then the system tries to find a better packet transmission rate. When it finds a better solution, all users adopt its α to the value. However, because of the punishment period T is not adjusted to an optimal value, the deviation can have benefits. So there exists a period that the utility and α switch from cooperation to noncooperation. In this period, T is increased until everybody realize that there is no benefit for deviation because of the long period of punishment. If the system is stable for time N, a new α is attemped to see if the performance can be improved. If yes, the new value is adopted, otherwise the original value is restored. So the packet forwarding probability is adjusted until the optimal solution is found, and the learned utility function is nondecreasing function. Notice that users are less reluctant to deviate when α is close to the optimal solution. This is because the benefit of deviation becomes smaller and users already have the estimated punishment time according to (7).

Figure 6 and Figure 7 show the packet forwarding probability and utility vs. normalized packet forwarding cost F/G for the optimal solutions and the solutions studied by the proposed framework, respectively. Here the system tries to find the new α for only 250 times. From the simulation results, we can see that the proposed framework can find the optimal packet forwarding probability and the optimal utility with maximum of 0.7% and 0.04% difference, respectively. This proves that the proposed framework can find the optimal packet forwarding probability very efficiently.

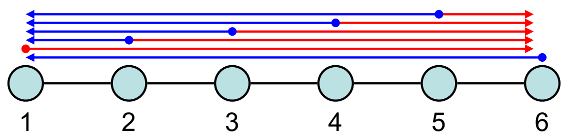

Fig. 8: Example of asymmetry network

B. Asymmetry Networks

Practical networks are generally asymmetry in nature. In this subsection, we turn our attention to analyze the performance of the proposed framework over the asymmetry networks. An example of synchronous asymmetry networks is shown in Figure 8. In this case, the node 1 and 6 act as the sinks of the information. The red arrows indicate the flows direction of routes where node 6 is the sink. In this case, node 1 to node 5 want to transmit to node 6. Similarly, the blue arrows indicate the flows direction where node 1 is the sink. In this case, node 2 to node 6 want to transmit to node 1. Notice that node 2 and node 3 are asymmetry. We formulate the utility function for node 1 to 6 as follows

$$U_1 = 2G\alpha_2\alpha_3\alpha_4\alpha_5$$
$$U_2 = G[1+\alpha_3\alpha_4\alpha_5] - F[2\alpha_2 + \alpha_2\alpha_3 + \alpha_2\alpha_3\alpha_4 + \alpha_2\alpha_3\alpha_4\alpha_5]$$
$$U_3 = G[\alpha_2 + \alpha_4\alpha_5] - F[2\alpha_3 + \alpha_2\alpha_3 + \alpha_3\alpha_4 + \alpha_3\alpha_4\alpha_5]$$
$$U_4 = G[\alpha_5 + \alpha_2\alpha_3] - F[2\alpha_4 + \alpha_3\alpha_4 + \alpha_4\alpha_5 + \alpha_2\alpha_3\alpha_4]$$
$$U_5 = G[1+\alpha_2\alpha_3\alpha_4] - F[2\alpha_5 + \alpha_4\alpha_5 + \alpha_3\alpha_4\alpha_5 + \alpha_2\alpha_3\alpha_4\alpha_5]$$
$$U_6 = 2G\alpha_2\alpha_3\alpha_4\alpha_5$$

We can see that the noncooperative solution for each node is to use zero forwarding probability. Notice that due to the symmetry in the network flow, node 2 and node 5, node 3 and node 4 have the same forwarding probability, respectively. Moreover node 1 utility and node 6 utility are totally depended on other nodes' packet forwarding probability. So the optimization parameters are α_2 ($\alpha_2 = \alpha_5$) and α_3 ($\alpha_3 = \alpha_4$) only. Since node 2 and node 4 have their own optimization goals, for system optimization point of view, this is a multiple objective optimization. To quantify the optimality, we need to define the following concept:

Definition 3: Pareto optimality is a measure of optimality. An outcome of a game is Pareto optimal if there is no other outcome that makes every node at least as well off and at least one node strictly better off. That is, a Pareto Optimal outcome cannot be improved upon without hurting at least one node. Often, a Nash Equilibrium is not Pareto Optimal implying that the players' payoffs can all be increased.

In Figure 9, we show the Pareto optimal region and the simulated results obtained by the proposed framework. The x-axis and y-axis are α_2 and α_3, respectively. Here the system tries to find new packet forwarding probability for 250 times. Any point within the shades area is Pareto optimal. Most of the simulated points are within this region. Very few points are located outside. This is due to the failure of searching the optimal packet forwarding probability within 250 times. We can see that the proposed framework is effective to find the Pareto optimum for asymmetry networks.

V. Conclusions

In this paper, we proposed a self-learning repeated game framework for packet forwarding networks. The cooperation within users for packet forwarding is obtained by threat of punishment in the future, while the optimal packet forwarding probability of each user can be studied distributively. From the simulation results for symmetry and asymmetry networks, we can see that the proposed framework can effectively find the solutions very close to the optimal solutions in a distributed way. The proposed framework can have impacts on the designs of future communication networks such as wireless networks, wired networks, Ad hoc networks, sensor networks, etc.

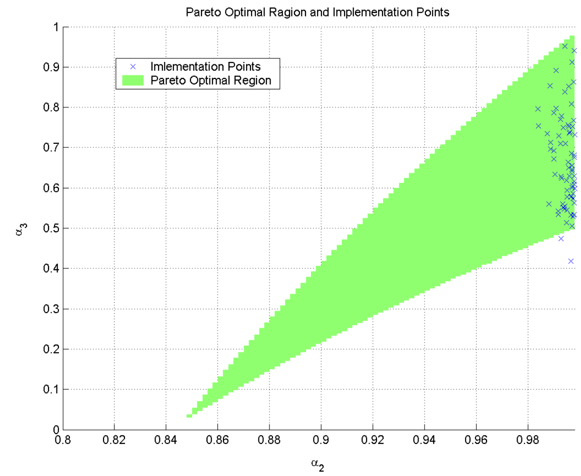

Fig. 9: Pareto optimal region and the simulated results

References

[1] G. Owen, "Game theory, 3rd ed." *Academic Press*, 2001.

[2] R. J. La and V. Anantharam, "Optimal routing control: repeated game approach", *IEEE Transactions on Automatic Control*, pp 437-450, March 2002.

[3] A. B. MacKenzie and S. B. Wicker, "Stability of multipacket slotted Aloha with selfish users and perfect information," *IEEE INFOCOM Conference*, 2003.

[4] A. B. MacKenzie and S. B. Wicker, "A repeated game approach to distributed power control in CDMA wireless data networks," *in revision for IEEE Transactions on Wireless Communications*.

[5] V. Srinivasan, P. Nuggehalli, C. F. Chiasserini and R. R. Rao "Cooperative in Wireless Ad Hoc Networks", *IEEE INFOCOM'03*, San Fransisco, March 30-April 03,2003.

[6] P. Michiardi and R. Molva "A Game Theoretical Approach to Evaluate Cooperation Enforcement Mechanisms in Mobile Ad hoc Networks", *IEEE/ACM Workshop WiOpt 2003*, Sophi Antipolis, France.

[7] P. Michiardi and R. Molva "CORE: A Collaborative Reputation Mechanism to enforce node cooperation in Mobile Ad hoc Networks", *IFIP-Communication and Multimedia Security Confrerence*, 2002.

[8] Eitan Altman, A. A. Kherani, P. Michiardi and R. Molva "Non-cooperative Forwarding in Ad-hoc Networks", *The 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2004).*

[9] M. Felegyhazi, L. Buttyan and J. P. Hubaux "Equilibrium Analysis of Packet Forwarding Stratiegies in Wireless Ad-hoc Networks - the static case", *Personal Wireless Communications 2003*, Venice, Italy.

[10] S. Buchegger and J-Y. Le Boudec "Performance Analysis of the CONFIDANT Protocol (Cooperation Of Nodes-Fairness in Dynamic Ad-hoc NeTworks", *In Proc. 3rd ACM Internation Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02)*, Lausannae, Switzerland,pp.80-91, June 9-11, 2002.

[11] Z. Han, Z. Ji, and K.J. Ray Liu, "Dynamic Distributed Rate Control for Wireless Networks by Optimal Cartel Maintenance Strategy", *IEEE Globecom04*.

[12] Z. Han, Z. Ji, and K. J. Ray Liu, "Low-complexity OFDMA Channel Allocation with Nash Bargaining Solution Fairness", *IEEE Globecom*, 2004.

[13] Z. Han, Z. Ji, and K. J. Ray Liu, "Power Minimization for Multi-cell OFDM Networks Using Distributed Non-cooperative Game Approach", *IEEE Globecom*, 2004.

